Q1
```
using System;
  class shape
  {
     public int H;
     public int W;
     public shape(int h, int w)
     {  H = h;
        W = w;
     }
     public double area()
     {  return 0;    }
  }
     class rectangle :shape
     { public rectangle(int p, int q) : base(p, q) { }
       public double area()
       {  return H * W;      }
     }
  class triangle:shape
  {  public triangle(int p, int q) : base(p, q) { }
       public double area()
       {  return (H * W)/2.0;      }
  }
  class Program
  {   static void Main(string[] args)
     {  shape s1 = new triangle(5, 5);
        Console.WriteLine (s1.area());
        shape s2 = new rectangle (5, 5);
        Console.WriteLine(s2.area());
        Console.ReadLine ();      }
  }
```
What will be the output

A. 0,0

B. 25,12

C. 12.00

D. None

Q2 .
```
using System;
        class shape
        {  public int H;
           public int W;
           public shape(int h, int w)
           {     H = h;
              W = w;    }
           public virtual double area()
           {      return 0;    }
        }
           class rectangle :shape
           { public rectangle(int p, int q) : base(p, q) { }
             public double area()
             {  return H * W;      }
           }
        class triangle:shape
        {
           public triangle(int p, int q) : base(p, q) { }
             public double area()
             {  return (H * W)/2.0;      }
```

```
}
class Program
{  static void Main(string[] args)
   {     shape s1 = new triangle(5, 5);
         Console.WriteLine (s1.area());
         shape s2 = new rectangle (5, 5);
         Console.WriteLine(s2.area());
         Console.ReadLine ();
   }
}
```

A. 25,12.00
B. 12.05,25
C. 0,0
D. None

Q3
```
using System;
using System.Collections.Generic;
   class shape
   {  public int H;
      public int W;
      public shape(int h, int w)
      {  H = h;
         W = w;     }
      public virtual double area()
      {        return 0;      }
   }
      class rectangle :shape
      { public rectangle(int p, int q) : base(p, q) { }
        public override double area()
        {  return H * W;      }
      }
   class triangle:shape
   {  public   triangle(int p, int q) : base(p, q) { }
      public override double area()
        {  return (H * W)/2.0;      }
   }
   class Program
   {   static void Main(string[] args)
      {   shape s1 = new rectangle(5, 5);
         Console.WriteLine (s1.area());
         shape s2 = new triangle(6, 6);
         Console.WriteLine(s2.area());
         Console.ReadLine ();
      }
   }
```

A. 0,25
B. 25 ,18
C. 25,18.000000
D. 0

Q4
```
using System;
namespace ConsoleApplication7
{
  abstract class shape
   {
```

```
    public int H;
    public int W;
    public shape(int h, int w)
    {   H = h;
        W = w;      }
    public virtual double area()
    {       return 0;      }
  }
  class rectangle :shape
    { public rectangle(int p, int q) : base(p, q) { }
      public  double area()
      {  return H * W;      }
    }
  class triangle:shape
  {
    public   triangle(int p, int q) : base(p, q) { }
    public  double area()
      {  return (H * W)/2.0;      }
  }
  class Program
  {
    static void Main(string[] args)
    {   shape s1 = new rectangle(5, 5);
        Console.WriteLine (s1.area());
        shape s2 = new triangle (6, 6);
        Console.WriteLine(s2.area());
        Console.ReadLine ();
    }
  }
```

A.**0,0**
B.25,18
c. 15,18.000000
d.none

Q5
```
using System;
using System.Collections.Generic;
namespace ConsoleApplication7
{
  abstract class shape
  {   public int H;
      public int W;
      public shape(int h, int w)
      {   H = h;
          W = w;    }
      public abstract double area();
  }
      class rectangle :shape
      { public rectangle(int p, int q) : base(p, q) { }
        public  override double area()
        {  return H * W;      }
      }
  class triangle:shape
  {   public   triangle(int p, int q) : base(p, q) { }
      public  override double area()
        {  return (H * W)/2.0;      }
  }
```

```
class Program
{ static void Main(string[] args)
  {  shape s1 = new rectangle (5, 5);
     Console.WriteLine (s1.area());
     shape s2 = new triangle (6, 6);
     Console.WriteLine(s2.area());
     Console.ReadLine ();
  }
}
```

A. 25,18
B. 0,0
C. 0,15
D. None

Q6 using System;

```
interface I1
{
    void A();
}

interface I2
{
    void A();
}

class C : I1, I2
{
    public void A()
    {
        Console.WriteLine("C.A()");
    }
}
Class entry
{ static void main(){
C c = new C();
I1 i1 = (I1)c;
I2 i2 = (I2)c;
}
c.A();
I1.A();
I2.A();
}
```

What will be the output of the program.

A. C.A()
C.A()
C.A()

```
B. c.A();
i2.A();
i1.A();

C.Error
D. None
```

Q7.using System;

```
interface I1
{ void A(); }

interface I2
{ void A(); }

class C : I1, I2
```

```
{  public void A()
    {  Console.WriteLine("C.A()");  }

    void I1.A()
    { Console.WriteLine("I1.A()");  }
}
Class entry
{ static void main(){

C c = new C();
c.A();
I2 i2 = c;
c.A();}}
```

A. C.A(),C.A()
B. C.A(),I1.A()
C I1.A(),C.A()
E. None

Q8 using System;

```
interface I1
{ void A();}

interface I2
{  void A();}

class C : I1, I2
{   void I1.A()
    { Console.WriteLine("I1.A()");  }
}
Class entry
{ static void main(){

C c = new C();
c.A();
```

}}
A.Compile time Error
B. I2.A()
C. Run time Error
d. none

Q9. using System;

```
interface I1
{   void A();}
interface I2
{ void A();}
class C : I1, I2
{    void I2.A()
    { Console.WriteLine("I2.A()");  }
    Void A() { Console.WriteLine("I1.A()");}
}
Class entry
{ static void main(){
C c = new C();
I2 x=new C();
c.A();
x.A();}}
```

A I1.A(),I2.A()
b. I2.A(),I1.A()
c.Error
d.None

Q 10 using System;

```
interface I1
{ void A();}

interface I2
{ void I1.A();}


class C : I1, I2
{   void I2.A()
    { Console.WriteLine("I2.A()"); }
   Void A() { Console.WriteLine("I1.A()");}
}
Class entry
{ static void main(){

I1 x=new C();
I2 p=new c();
x.A();
p.A();
}}
```

A. **I1.A(),I2.A()**
B. I1.A();
C. Error
D. None

Q11

```
using System;
class Test
{ static void Main() {
      A.F();
      B.F()   }
}
class A
{ static A() {
      Console.WriteLine("Init A");
   }
   public static void F() {
      Console.WriteLine("A.F");
   }
}
class B
{  static B() {
    Console.WriteLine("Init B");
   }
   public static void F() {
      Console.WriteLine("B.F");
   }
}
```

What will be the output

A.

```
Init A
A.F
Init B
B.F
```

B.None

C.
```
Init B
B.F
Init A
A.F
```

D.Error

Q12 Readonly variable must be either initialised at time of declaration or in constructor

A.True

B. False

Q13 Readonly variable must be initialised at time of declaration

A.true

B. **false**

Q14 Readonly variable required data at compile time

A. True;

B. False

Q15.Const variable required data at runtime

a. True

b. False

Q16 class program

```
{ const int a=5;
   Static void Main()
   { program C= new program();
  Console.WriteLine(C.a);
    }
}
```

A. Compile time Error

b.5

c. none

d. run time error

Q17

```
class program
{ const int a=5;
   Static void main()
   { program C= new program();
  Console.WriteLine(program.a);
    }
}
```

a. Compile time Error

b.5

c. none

d. run time error

Q18

You can not declare static variable inside method

a.True

b.False

Q19 You can not declare constant variable inside method

a.true

b.false

Q20 constant variable are by default static

a. True

b. false